

PERANCANGAN SISTEM PAKAR UNTUK DIAGNOSA PENYAKIT MULUT DAN GIGI MENGUNAKAN BAHASA PEMROGRAMAN CLIPS

I Nyoman Kusuma Wardana, Alfa Antariksa, Nazrul Effendy

Jurusan Teknik Fisika, Fakultas Teknik

Universitas Gadjah mada

Jl. Grafika No. 2 Yogyakarta 55281

Telp : (0274)902120, fax : (0274)580882

E-mail: koes_wrd@yahoo.com, alfa_cumlaude@yahoo.co.id, nazrul@gadjahmada.edu

ABSTRAKSI

Sistem pakar sebagai sistem yang mengadopsi kepakaran manusia ke dalam komputer memiliki kedudukan strategis sebagai sistem yang dapat membantu menyelesaikan permasalahan di bidang kedokteran. Sulitnya menentukan jenis penyakit karena rumitnya berbagai gejala yang mengiringinya, dapat dibantu dengan merepresentasikan gejala suatu penyakit ke dalam suatu bahasa pemrograman komputer. Pada penelitian ini, dikembangkan suatu metode untuk mendignosa 18 jenis penyakit mulut dan gigi berdasarkan 40 jenis gejala yang mengiringinya. Lingkungan pengembangan sistem pakar pada penelitian ini menggunakan bahasa pemrograman CLIPS.

Kata kunci: sistem pakar, penyakit mulut dan gigi, CLIPS.

1. PENDAHULUAN

Penyakit mulut dan gigi dapat ditentukan dengan melihat berbagai gejala yang mengiringi penyakit tersebut. Mengalihkan pengetahuan seorang pakar penyakit mulut dan gigi ke dalam komputer tentunya akan sangat membantu dokter untuk mendiagnosa penyakit pasien.

Pada penelitian ini, gejala yang mengiringi penyakit mulut dan gigi dirangkum menjadi 40 jenis. Kumpulan gejala ini digunakan untuk membedakan 18 jenis penyakit mulut dan gigi.

Penelitian ini membahas penerapan sistem pakar melalui pemrograman perangkat lunak. Berbagai fakta dan aturan untuk setiap jenis penyakit ditanamkan pada program agar mampu menentukan jenis penyakit yang cocok sesuai dengan gejalanya. CLIPS (*C Language Integrated Production System*) dipilih sebagai lingkungan pengembangan sistem pakar.

Sistem pakar dapat dibangun dengan berbagai perangkat lunak seperti CLIPS, Prolog, LISP dan berbagai perangkat lunak yang lain. CLIPS memiliki fasilitas yang lengkap untuk membangun suatu sistem pakar karena memiliki berbagai fitur seperti editor yang terintegrasi dan *debugging tool* [2].

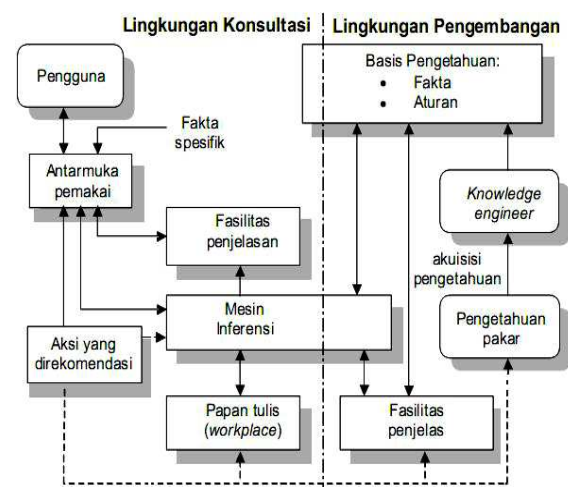
2. DASAR TEORI

2.1 Sistem Pakar

Kepakaran (*expertise*) adalah adalah pengetahuan yang ekstensif (meluas) yang diperoleh melalui rangkaian pelatihan, membaca dan pengalaman [1]. Pembentukan sistem pakar didasarkan pada suatu ide untuk mentransfer

pengetahuan seorang pakar (atau sumber kepakaran yang lain) ke dalam komputer. Pengetahuan yang tersimpan ini selanjutnya digunakan untuk menyelesaikan permasalahan yang sesuai dengan bidang kepakaran tertentu. Peran sistem pakar dewasa ini semakin dirasa penting untuk menyelesaikan permasalahan diberbagai bidang, termasuk bidang kesehatan.

Sistem pakar terdiri dari dua bagian pokok, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation enviromnet*). Pembentukan basis aturan dan pembangunan komponen dilakukan pada lingkungan pengembangan, sedangkan lingkungan konsultasi digunakan sebagai sistem konsultasi oleh orang yang bukan ahli [5].



Gambar 1. Struktur sistem pakar [1]

Berdasarkan pengalaman, kegagalan pembangunan sebuah sistem pakar disebabkan para ahli sangat susah untuk menyatakan secara tepat pengetahuan dan aturan yang digunakan untuk menyelesaikan masalah [6]. Agar menjadi efektif, sistem pakar harus memuat sejumlah substansi kepakaran yang terorganisir [4].

2.2 Bahasa Pemrograman CLIPS

CLIPS (*C Language Integrated Production System*) adalah program *expert system* yang pertama kali di *release* tahun 1986 dan dikembangkan oleh Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center. Sejak perama kali dirilis, CLIPS menunjukkan perkembangan yang pesat, dan saat ini digunakan oleh ribuan orang untuk mengembangkan sistem pakar diseluruh dunia [2].

Terdapat tiga cara untuk merepresentasikan pengetahuan pada CLIPS yaitu [2]:

1. *Rules*, dirancang untuk pengetahuan heuristik yang berbasiskan pengalaman.
2. *Deffunction* dan *generic function*, dirancang untuk pengetahuan prosedural.
3. *Object-oriented programming*, juga dirancang untuk pengetahuan prosedural yang mendukung : *classes*, *message-handlers*, *abstraction*, *encapsulation*, *inheritance*, dan *polymorphism*. Aturan (*rule*) dapat mencocokkan antara objek dan fakta.

Sebagai perangkat pengembangan sistem pakar, CLIPS dilengkapi fitur yang lengkap mengenai elemen dasar sistem pakar, meliputi [2]:

1. *Fact-list*, dan *instance-list*, memuat berbagai data
2. *Knowledge-base*, memuat *data base* aturan
3. *inference engine*, mengontrol semua eksekusi aturan (kapan dan aturan mana yang harus di eksekusi)

Penulisan program pada CLIPS dapat berupa serangkaian aturan (*rules*), fakta (*facts*) dan objek (*objects*). Namun CLIPS mempunyai kemampuan merepresentasikan walaupun hanya berupa aturan dan fakta.

3. METODE PENELITIAN

Penelitian diawali dengan pengumpulan fakta-fakta mengenai penyakit dan gejalanya. Berdasarkan pengumpulan tersebut, dirangkum penyakit dan gejala yang menyertai penyakit mulut dan gigi pada Tabel 1 dan Tabel 2.

Tabel 1. Data Nama Penyakit Mulut dan Gigi

No	Nama Penyakit
1	Trench Mouth
2	Abses Periapikal

No	Nama Penyakit
3	Gingivitis simplek
4	Gingivitis herpetik akut
5	Periodontitis
6	Pulpitis
7	Sariawan
8	Kanker dan Tumor
9	Infeksi Herpes
10	Bau mulut
11	Canker
12	Angina ludwig
13	Liken planus
14	Keilitis
15	Pellagra
16	Mslh kelenjar ludah
17	Eritema multiformis
18	Kelainan Sendi Temporomandibuler

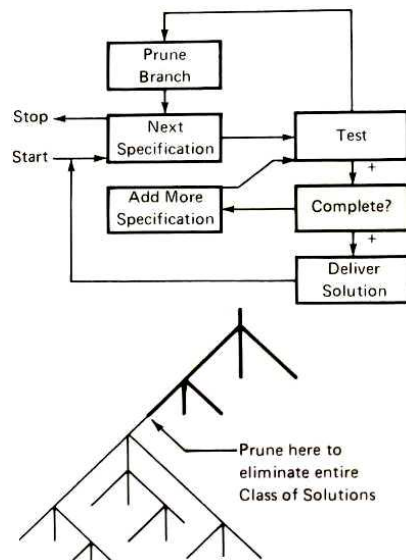
Tabel 2. Data Gejala Penyakit Mulut dan Gigi

No	Gejala
1	Bau mulut busuk
2	Bibir pecah-pecah
3	Disertai sakit kepala
4	Endapan plak pada karang gigi
5	Gusi berwarna merah
6	Gusi membengkak
7	Gusi mudah berdarah
8	Lidah kemerahan
9	Lidah menyumbat saluran pernapasan
10	Lidah terasa licin
11	Luka pada mulut
12	Menggertakkan gigi
13	Nanah pada pangkal gusi
14	Nyeri pada lidah
15	Nyeri pada otot pengunyah
16	Nyeri pada gusi
17	Nyeri saat mengunyah dan menelan
18	Pembengkakan di daerah langit-langit lunak
19	Pembengkakan di dasar mulut
20	Pembengkakan klnjar getah bening di leher
21	Pembengkakan kelenjar ludah
22	Pembusukan gigi
23	Peradangan pada pulpa
24	Pipi dalam terasa gatal
25	Stress/Gelisah
26	Sudut mulut terasa nyeri
27	Terbentuk kantong diantara gigi dan gusi
28	Terbentuk kista di dasar mulut
29	Trdpat benjolan putih pd pipi bagian dalam
30	Trdpat bercak putih berlendir pada mulut
31	Terdapat bercak putih pada mukosa mulut
32	Terdapat luka bakar
33	Terdapat ulkus pada mukosa
34	Terjadi daerah kemerahan di mulut
35	Terjadi demam
36	Terjadi infeksi pada kelenjar ludah

No	Gejala
37	Tidak enak badan
38	Tubuh mengalami kelelahan
39	Ujung gusi tertutup slaput berwarna abu-abu
40	Ujung-ujung gusi mengalami pengikisan

Pemecahan masalah (*problem-solving*) pada sistem ini menggunakan formulasi *generate-and-test*, dan sistem penyimpulan menggunakan metode *forward chaining*. Alur kerja program terlihat seperti suatu sistem hirarki.

Formula *generate-and-test* dibagi menjadi dua bagian: *generator* dan *tester* [3]. *Generator* membangkitkan pemecahan yang mungkin dan *tester* akan memotong solusi dalam cabang hirarki yang tidak memungkinkan menjadi solusi pemecahan. Hal ini diseleksi dari jawaban “y” dan “t” (masing-masing menyatakan “ya” dan “tidak”) yang menjadi jawaban setiap pertanyaan.



Gambar 2. Hirarki formula *generate-and-test* [3]

Program mengambil keputusan berdasarkan aturan-aturan yang dibangkitkan dari setiap pertanyaan yang diajukan pada pasien. Berikut penggalan program untuk mengumpulkan data:

Algoritma 1. Penggalan program pengumpulan data

```
;;;=====PENGUMPULAN DATA GEJALA=====

(defrule nyerigusi-1
  (observasi)
  =>
  printout t "Apakah terasa nyeri pada
  gusi?(y/t)"
  (assert(gejala enambelas (read)))
  (printout t crlf)
)
;;;=====16-ya

(defrule bengkakelenjar-1
```

```
(gejala enambelas y)
=>
(printout t "Terdapat pembengkakan
kelenjar getah bening")
(printout t " di leher?(y/t)")
(assert(gejala duapuluh (read)))
(printout t crlf)
)
;;;=====20-ya
```

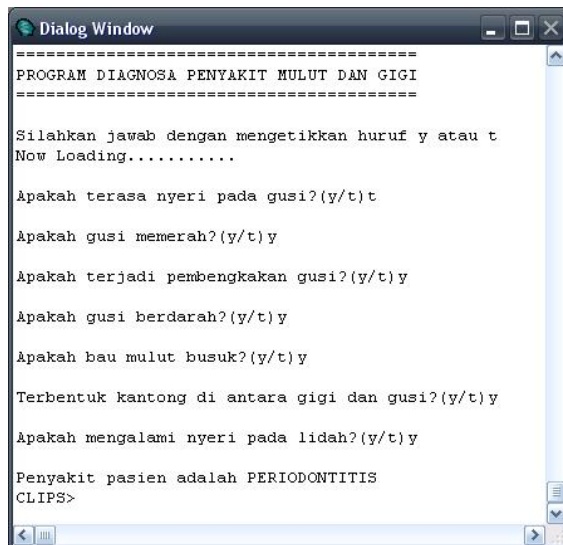
Ketika kumpulan data memenuhi ciri suatu penyakit tertentu, maka program akan mengeluarkan hasil diagnosa. Setiap jawaban yang berbeda dari setiap pertanyaan akan menghasilkan fakta yang berbeda dan pertanyaan yang dikeluarkan berikutnya juga berbeda karena penerapan sistem *generate-and-test*. Pada algoritma 2 terlihat bahwa penyakit *trench mount* akan terdiagnosa ketika terjadi kombinasi jawaban dari delapan gejala yang ditanyakan.

Algoritma 2. Penggalan Kumpulan Aturan Program

```
;;;=====SOLUSI PENYAKIT=====
;;;=====1
(defrule trenchmounth
  (gejala enambelas y)
  (gejala duapuluh t)
  (gejala sebelas t)
  (gejala duadua t)
  (gejala tujuhbelas a y)
  (gejala duapuluh a t)
  (gejala empatpuluh y)
  (gejala tigasembilan y)
  =>
  (printout t "Penyakit pasien adalah")
  (printout t "TRENCH MOUNTH" crlf)
)
```

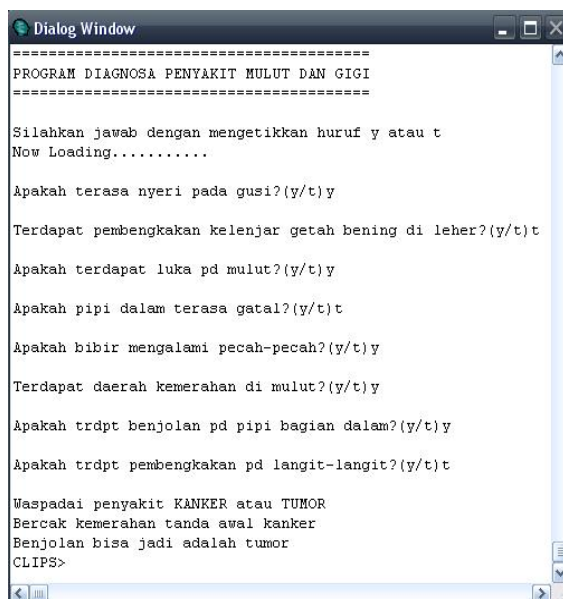
4. HASIL DAN PEMBAHASAN

Tampilan anatarmuka komputer dan *user* terlihat seperti pada gambar 3. setiap pasien dihadapkan pada pertanyaan yang mengharuskan untuk menjawab dengan menuliskan “y” atau “t”.



Gambar 3. Tampilan antarmuka komputer dan user

Tampilan pada gambar 3 memperlihatkan hasil diagnosa spesifik yang menunjukkan ciri-ciri penyakit periodontitis. Namun, terkadang terdapat gejala yang tidak spesifik yang merujuk pada suatu penyakit tertentu. dalam kondisi ini, proram tetap merespon dengan mengeluarkan hasil diagnosa dengan keterangan tambahan tertentu. Keadaan seperti ini terlihat pada gambar 4.



Gambar 4. Diagnosa dengan tambahan keterangan

Dalam keadaan tertentu, program juga dapat memberi respon bahwa gejala di luar data base komputer. Hal ini sangat tergantung pada tahap pemeliharaan program selanjutnya. Penambahan dan update sistem menjadi mutlak diperlukan, untuk menghadapi dinamika penyakit dan gejala yang mengiringinya. Gambar 5 menunjukkan kondisi di atas.



Gambar 5. Diagnosa di luar data base komputer

5. KESIMPULAN

Program pembuatan sistem pakar ini bertujuan untuk membantu user mendiagnosa penyakit mulut dan gigi. Sistem pakar ini tidak dapat 100% dijadikan sebagai *final decision* dalam menentukan penyakit yang dialami pasien. Penalaran yang diperoleh dari pengalaman yang dimiliki oleh user tetap menjadi faktor utama dalam sistem diagnosa penyakit mulut dan gigi. Namun, hasil program ini akan berusaha mengarahkan user untuk fokus terhadap penyakit yang dialami pasien berdasarkan gejala yang ditimbulkan.

6. PUSTAKA

- [1] Achmad, Balza, *Diktat Mata Kuliah Kecerdasan Buatan*, Jurusan Teknik Fisika UGM, Yogyakarta, 2006.
- [2] Giarratano, J. C., *CLIPS User Guide*, <http://www.ghg.net/clips/download/documentation/>.
- [3] Hayes-Roth, F., Waterman, D.A., Lenat, D.B., *Building Expert Systems*, Addison-Wesley Publishing Company, Inc., Massachusetts, 1983.
- [4] Klahr, P., Waterman, D.A., *Expert Systems: Techniques, Tools and Applications*, The Rand Corporation, Canada, 1986.
- [5] Kusumadewi, S., *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta, 2003.
- [6] Suyoto, *Intelegensi Buatan: Teori dan Pemrograman*, Gava Media, Yogyakarta, 2004.
- [7] _____, <http://www.medicastrore.com/>.